

Vysoká škola báňská – Technická univerzita Ostrava

Fakulta strojní

Katedra automatizační techniky a řízení

Automatizovaný závlahový systém

Self-watering system

Vypracoval:

Tomáš Repka

Vedoucí bakalářské práce:

Ing. Pavel Smutný, Ph.D.

Ostrava 2016

Zadání bakalářské práce

Student: **Tomáš Repka**
Studijní program: B2341 Strojírenství
Studijní obor: 3902R001 Aplikovaná informatika a řízení
Téma: Automatizovaný závlahový systém
Self-watering system
Jazyk vypracování: čeština

Zásady pro vypracování:

1. Seznamte se a popište vývojovou desku Arduino.
2. Popište možnosti automatizovaných závlahových systémů.
3. Proveďte analýzu vhodných hardwarových prostředků.
4. Navrhněte a vytvořte vlastní model automatizovaného závlahového systému.
5. Zhodnoťte dosažené výsledky a navrhněte směry dalšího řešení.

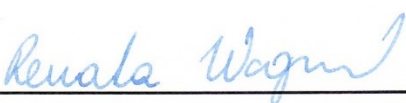
Seznam doporučené odborné literatury:

MARGOLIS, Michael. Arduino cookbook. 2nd ed. Sebastopol, Calif.: O'Reilly, c2012, 699 p. ISBN 1449313876.
MELGAR, Enrique Ramos, Ciriaco CASTRO DÍEZ a Przemek JAWORSKI. Arduino and Kinect projects: Design, build, blow their minds. New York: Distributed to the book trade worldwide by Springer Science+Business Media, c2012, 393 p. ISBN 9781430241683.
IGOE, Tom. Making things talk. 2nd ed. Sebastopol: O'Reilly, 2011, xvi, 470 s. ISBN 978-1-449-39243-7.
Arduino.cc [online]. 2015 [cit. 2015-10-07]. Dostupné z: <https://www.arduino.cc/>
Arduino.cz [online]. 2015 [cit. 2015-10-07]. Dostupné z: <http://arduino.cz/>


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Pavel Smutný, Ph.D.**

Datum zadání: 11.12.2015
Datum odevzdání: 16.05.2016


doc. Ing. Renata Wagnerová, Ph.D.
vedoucí katedry

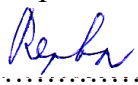



doc. Ing. Ivo Hlavatý, Ph.D.
děkan fakulty

Místopřísežné prohlášení studenta

Prohlašuji, že jsem celou bakalářskou práci včetně příloh vypracoval samostatně pod vedením vedoucího bakalářské práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě: 16.05.2016


.....

Tomáš Repka

Prohlašuji, že

- jsem byl seznámen s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo.
- беру на ве́домі, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo nevýdělečně ke své vnitřní potřebě bakalářskou práci užít (§ 35 odst. 3).
- souhlasím s tím, že bakalářská práce bude v elektronické podobě uložena v Ústřední knihovně VŠB-TUO k nahlédnutí a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že údaje o kvalifikační práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své díl – bakalářskou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- беру на ве́домі, že odevzdáním své práce souhlasím se zveřejněním své práce dle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě: 16.05.2016



Tomáš Repka

Tomáš Repka

Obecní 129/4

747 73 Opava - Vávrovice

Anotace

REPKA, T. Automatizovaný závlahový systém. Ostrava: VŠB – Technická univerzita Ostrava, Fakulta strojní, Katedra automatizační techniky a řízení, 2016, 41 s. Bakalářská práce, vedoucí: Ing. Pavel Smutný, Ph.D.

Tato bakalářská práce se zabývá návrhem automatizovaného zavlažovacího systému. První část obsahuje seznámení s řídicí jednotkou – vývojovou deskou Arduino, způsoby různých zapojení komponent, seznámení se způsoby a možnostmi zavlažování. Dále návrhem vhodných komponent a návrhem konstrukce zavlažovacího systému rostliny. Nakonec je popsána samotná realizace systému včetně postupu při vytvoření řídicího programu.

Klíčová slova: zavlažování, automatizace, Arduino

Annotation

REPKA, T. Self-watering system. Ostrava: VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Control Systems and Instrumentation, 2016, 41 p. Bachelor thesis, thesis head: Ing. Pavel Smutný, Ph.D.

This bachelor thesis describes the design of an automated irrigation system. The first part of the document includes description of the control unit – Arduino development board, different ways of wiring components and possibilities of irrigation. Next part is a selection of appropriate components and design of construction of the irrigation system. At the end is described creation of the system and control software.

Key words: irrigation, automation, Arduino

Obsah

SEZNAM POUŽITÝCH ZKRATEK	7
ÚVOD	9
1 SEZNÁMENÍ S VÝVOJOVOU DESKOU ARDUINO	10
1.1 Verze Arduina	10
1.2 Moduly a senzory pro Arduino	11
1.3 Popis vývojové desky Arduino UNO	13
1.4 Arduino IDE a programování Arduina v knihovně Wiring	15
2 MOŽNOSTI AUTOMATIZOVANÝCH ZÁVLAHOVÝCH SYSTÉMŮ ...	16
2.1 Způsoby zavlažování	17
2.2 Popis fungování zavlažovacího systému	17
3 VÝBĚR KOMPONENT	18
4 NÁVRH SYSTÉMU	23
4.1 Možné technické problémy	23
4.2 Schéma zapojení systému	24
4.3 Vytvoření systému	25
4.4 Připojení komponent k Arduinu	26
5 VYTVÁŘENÍ PROGRAMU	28
5.1 Nastavení IDE	28
5.2 Vložení knihoven a definice proměnných	29
5.3 Program pro zavlažování	30
5.4 Připojení ESP8266 k WiFi síti	33
5.5 Posílání dat na ThingSpeak	35
6 ZÁVĚR	37
7 POUŽITÁ LITERATURA.....	40

Seznam použitých zkratek

802.11 b/g/n – standard pro bezdrátovou síť

ARM – Advanced RISC Machine, označení architektury procesorů s nízkou spotřebou elektrické energie

C – nízko úroňový programovací jazyk

C++ - programovací jazyk s podporou několika programovacích stylů

DC – Direct Current, stejnosměrný proud

FLASH – elektricky programovatelná paměť

GPRS – General Packet Radio Service, služba pro přenos dat

GPS – Global Positioning System, systém pro přesné určení polohy a času na Zemi

GSM – Global System for Mobile Communications, standard sítě pro mobilní telefony

I/O – Input/Output, vstup a výstup

I²C – Inter-Integrated Circuit, sériová sběrnice pro nízkorychlostní periférie

ICSP – In-Circuit Serial Programming, protokol pro sériové programování

IDE – Integrated Development Environment, integrované vývojové prostředí pro Arduino

IP – Internet Protocol, adresa pro identifikování síťového rozhraní

LCD – Liquid Crystal Display, zobrazovací zařízení, skládající se z pixelů

LED – Light-Emitting Diode, polovodičová elektronická součástka, která vyzařuje světlo

MISO – Master In Slave Out, Slave linka pro odesílání dat na Master linku

MOSI – Master Out Slave In, Master linka pro odesílání dat do periférií

RAM – Random-Access memory, počítačová paměť s přímým přístupem

RJ-45 – typ zapojení síťových kabelů

RTC – Real Time Clock, modul reálného času

Rx – Receive, linka pro příjem dat

SCK,SCL – Serial Clock, hodinové pulsy, sloužící pro synchronizaci přenosu dat

SD – Secure Digital, typ paměťových flash karet

SDA – Synchronous Data, datový kanál

SIM – Subscriber Identity Module, identifikační karta pro identifikaci v mobilní síti

SMS – Short Message Service, služba pro zasílání krátkých textových zpráv

SPI – Serial Peripheral Interface, sériové periferní rozhraní

SS – Slave Select, pin pro určení zařízení pro komunikaci

TCP – Transmission Control Protocol, protokol transportní vrstvy v síti Internet

Tx – Transmit, linka pro odesílání dat

USB – Universal Serial Bus, univerzální sériová sběrnice používaná pro připojení periférií

USB OTG – USB On-The-Go, specifikace, která umožňuje komunikovat jako hostitelské zařízení pro další USB zařízení

WiFi – Wireless Fidelity, bezdrátová komunikace v počítačových sítích

Úvod

V dnešní době se čím dál tím více používá pojem internet věcí, což znamená propojení vestavěných zařízení s internetem. Díky tomu, že dnes už má skoro každý chytrý telefon a připojení na síť internet, je propojování a ovládání zařízení ještě jednodušší. Můžeme si tak vytvořit například inteligentní domácnost, která nám bude pomáhat při běžných domácích činnostech a zvýší komfort našeho bydlení.

Jako příklad takových zařízení můžeme uvést robotické vysavače, které se samostatně pohybují po bytě a automaticky se vyhýbají překážkám. Dále třeba inteligentní osvětlení a žaluzie, které se nám postarají o přísun světla v době, kdy jej opravdu potřebujeme.

Všechna tato zařízení se dají koupit v obchodech a existují i firmy, které se specializují na vytvoření inteligentní domácnosti. Pořizovací ceny ale nejsou nejmenší, a pokud bychom měli omezený rozpočet, můžeme si sestavit svou vlastní inteligentní domácnost s využitím možností open hardware, což je hardware, který můžeme libovolně upravovat. Příkladem je elektronická deska Arduino, která do této skupiny patří.

Cílem bakalářské práce je vytvoření automatizovaného zavlažovacího systému pomocí vývojové elektronické desky Arduino. Na začátku se seznámíme se samotnou deskou Arduino, jejími doplňky a dalším materiálem, který bude zapotřebí pro realizaci zavlažovacího systému. Poté budu pokračovat v seznámení s integrovaným vývojovým prostředím Arduina, které se používá pro jeho programování. Dále popíši svůj návrh systému a zapojení jednotlivých komponent. Seznámíme se s možnými problémy a možnostmi jak se jich vyvarovat. Nakonec popíši fyzickou realizaci systému včetně vytvoření programu pro jeho ovládání a ukládání a zpracování dat.

1 Seznámení s vývojovou deskou Arduino

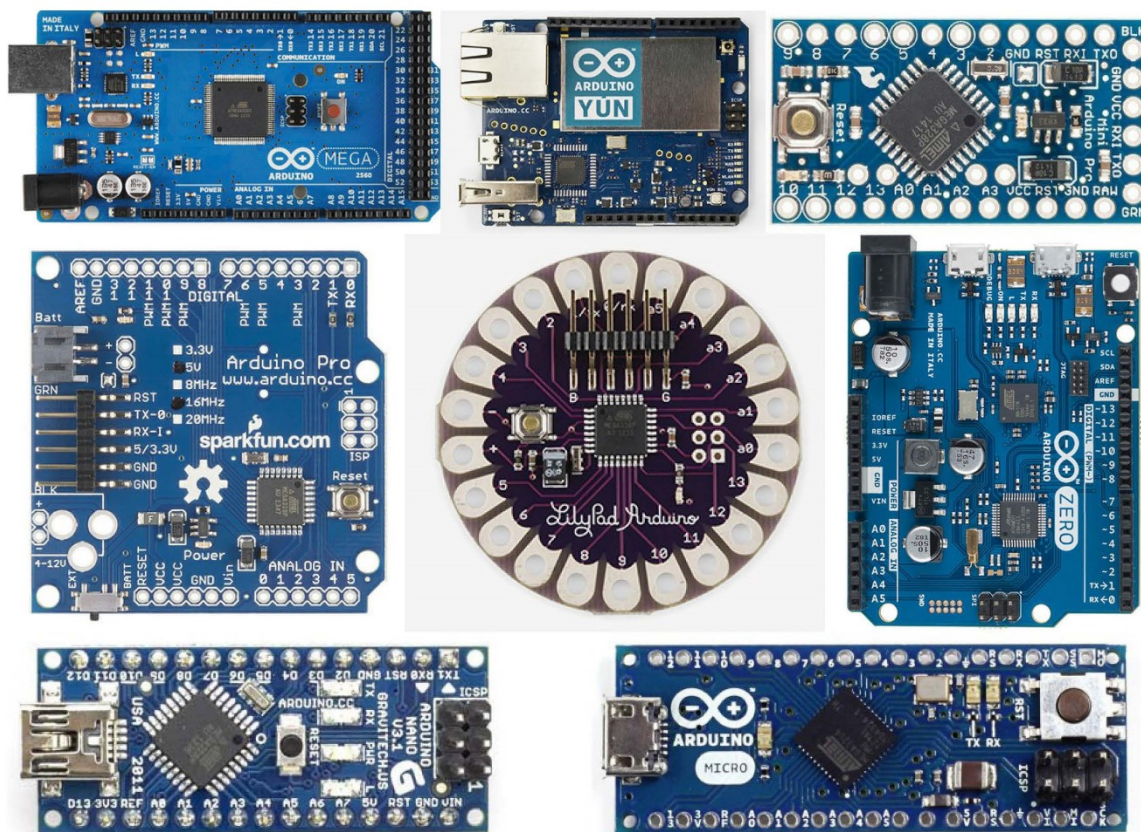
Arduino je programovatelná elektronická deska s vlastním procesorem a dalšími volitelnými elektronickými komponenty. Je to otevřená (open source) platforma, tudíž se dá na webu najít mnoho článků, návodů a hotových programů a také se vyrábí desítky připojitelných hardware příslušenství (například pohybové senzory a moduly). Díky tomu je Arduino vhodné pro všechny příznivce moderních technologií, kteří mají zájem o vytvoření vlastního systému, nebo třeba jen o pochopení jak dnešní systémy fungují.

1.1 Verze Arduina

Samotná deska Arduino má více verzí, lišících se například rozměry, použitými komponenty nebo porty, výkonem a také cenou. Nejrozšířenější verzí je Arduino UNO, kterou využijí ve svém zavlažovacím systému, a které si podrobněji popíšeme později.

Další verzí je Arduino **PRO MINI**, které patří mezi nejmenší desky a proto se využívá hlavně v malých přenosných zařízeních, nebo ovladačích. Tato verze používá procesor ATmega328 a neobsahuje USB konektor a je tedy nutné pro programování pomocí USB použít serial převodník. Verze Arduino **NANO** je o něco větší než předchozí verze, protože obsahuje USB port. Má o dva analogové vstupy více, jinak se moc od sebe neliší. Arduino **PRO** existuje ve dvou provedeních, a to 3,3 V / 8 MHz a 5 V / 16 MHz, jinak má podobné specifikace jako Arduino UNO, jen neobsahuje USB port a letovací piny. Další Arduino **MEGA** pracuje s procesorem ATmega2560. Je větší než UNO a má více vstupů a výstupů. Podobná verze je Arduino DUE, která jako první využívá 32-bitový ARM procesor. Podporuje například funkci USB OTG, díky které je možno připojit a pracovat s různými komponenty. Vzhledově odlišné Arduino **LILIPAD** je přizpůsobeno k nošení na oblečení. Spoje jsou provedeny vodivou nití a je tak možno například rozblikat určité části textilu. Existuje více druhů této desky, lišící se přítomností USB portu. Arduino **YUN** se liší od ostatních verzí tím, že používá procesor ATmega32U4, jako i další desky, ale také používá procesor AR9331, na kterém můžeme zprovoznit jednodušší distribuci linuxu. Tyto dva procesory mohou mezi sebou pomocí převodníku, mostu, komunikovat. Obsahuje také ethernet a WiFi modul, což se hodí pro připojení k síti a internetu. Další z mnoha verzí je Arduino **ZERO**, které je podobné jako Arduino UNO, ale používá 32-bitový procesor

ATSAMD21G18, tudíž je mnohem rychlejší a výkonnější. Arduino **MICRO** obsahuje procesor ATmega32U4, který obsahuje vše potřebné pro to, aby se po připojení k počítači mohl tvářit jako klávesnice, nebo myš a posílat příkazy typu stisknutí klávesy, nebo pohybu myši. Díky tomu je jednoduché vytvořit si vlastní klávesnici, nebo nějaký ovladač k počítači. [Arduino.cc, 2016]



Obrázek 1.1: Jednotlivé verze desek Arduino [Arduino.cc, 2016]

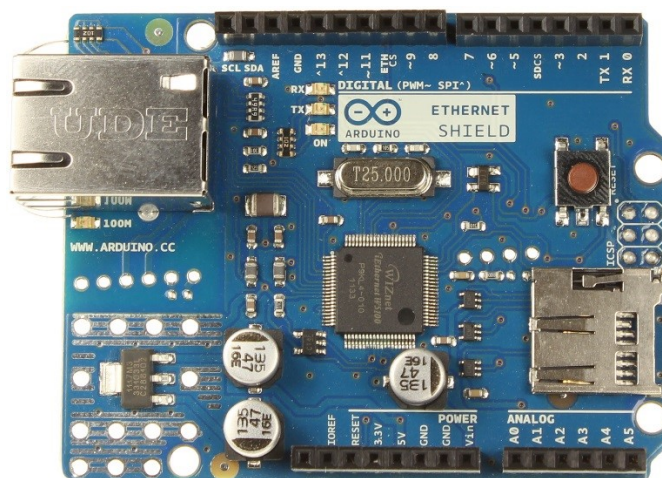
1.2 Moduly a senzory pro Arduino

Jak již bylo napsáno, existují desítky senzorů, které můžeme připojit a používat s deskou Arduino. Z těch nejjednodušších třeba teploměr, ať už digitální, nebo analogový. Dále třeba ultrazvukový měřič vzdálenosti, měřič tlaku, měřič intenzity světla a pohybové čidlo. Můžeme připojit také různé displeje, od obyčejných jednobarevných až po barevné dotykové displeje. Také se dá připojit GPS modul, pro zjišťování polohy.

Pokud nám na Arduinu nějaká funkce schází, je možno desku doplnit o modul - shield, který požadovanou funkci doplní. Těchto modulů je celá řada a je pouze potřeba zjistit, zda

je vybraný modul kompatibilní s deskou Arduino kterou vlastníme. Poté se modul pouze zasune do původní desky Arduino.

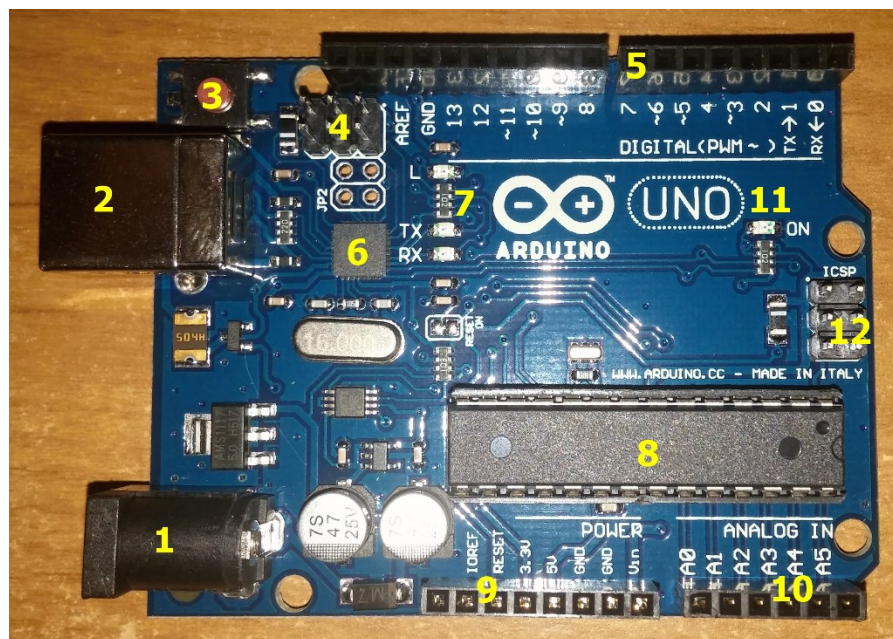
Existuje například Ethernet shield (Obrázek 1.2), který nám umožní připojit Arduino do sítě pomocí standartního RJ-45 konektoru. Dále třeba GSM shield, který zvládne připojit Arduino k internetu pomocí GPRS. Stačí pouze připojit SIM kartu s GPRS pokrytím. Také se dá telefonovat nebo posílat SMS zprávy. Pro další připojení k internetu můžeme využít také WiFi shield, který také obsahuje slot pro micro-SD kartu. Arduino Motor shield slouží pro řízení zátěží, například relé, DC a krokové motory, solenoidy. [Arduino.cc, 2016]



Obrázek 1.2: Arduino Ethernet Shield [Arduino.cz, 2016]

1.3 Popis vývojové desky Arduino UNO

Arduino UNO využívá procesor ATmega328P a jako první obsahuje USB port, který se využívá pro nahrávání programů. Existuje mnoho neoriginálních kopií a klonů, které mají většinou stejnou funkčnost, jen se musí ručně instalovat ovladač pro připojení k počítači.



Obrázek 1.3: Vývojová deska Arduino UNO

- Procesor: ATmega328P
- Operační napětí: 5 V
- Vstupní napětí: doporučeno 7-12 V, limit je 6-20 V
- Digitální I/O piny: 14
- Analogové vstupní piny: 6
- Flash paměť: 32 kB
- SRAM: 2 kB
- EEPROM: 1 kB
- Frekvence jádra procesoru: 16 MHz
- Délka: 68,6 mm
- Šířka: 53,4 mm
- Hmotnost: 25 g

[Arduino.cc, 2016]

Na desce Arduina (Obrázek 1.3) se vyskytuje několik komponent, mezi které patří:

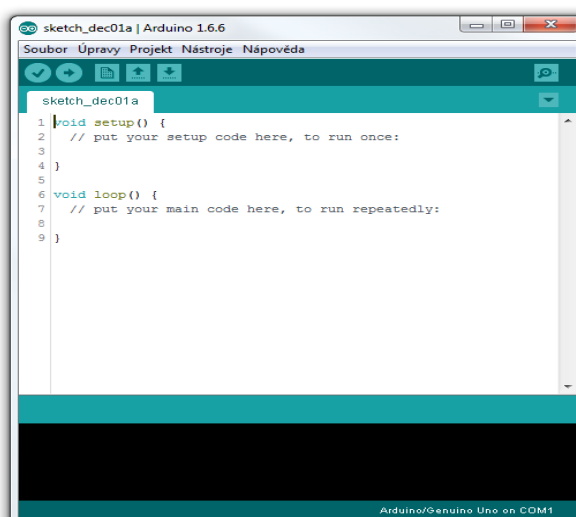
1. Napájecí konektor. Arduino můžeme napájet buď pomocí USB kabelu, nebo pomocí tohoto konektoru pro externí zdroj.
2. USB konektor typu B. Používá se pro připojení Arduina k počítači a nahrávání programu, funguje i jako napájení.
3. Tlačítko RESET. Použijeme jej pro restartování programu.
4. ICSP hlavice. Pomocí níž se dá externě programovat USB-serial převodník.
5. Digitální piny. Slouží pro komunikaci s připojeným zařízením. Komunikace probíhá číslicovou informací. Piny, které jsou označené vlnovkou, zvládnou i pulzně šířkovou modulaci. Piny 0 a 1 se používají také jako piny Rx (*receive* – přijmout) a Tx (*transmit* – odeslat), které slouží pro sériovou komunikaci, například pro spojení dvou desek Arduino. Piny 10, 11, 12, 13 se také využívají jako SS, MOSI, MISO, SCK piny pro sběrnici SPI – sériové periferní rozhraní. Ta se využívá pro tzv. full-duplex připojení, to znamená, že zařízení může komunikovat v obou směrech (přijímat i odesílat data) ve stejný čas. Nevýhodou je narůstající počet vodičů s každým přidaným zařízením.
6. USB-serial převodník. Slouží ke komunikaci mezi čipem a počítačem.
7. LED diody L, Rx, Tx. Diodu L můžeme využít místo externí LED diody, je totiž připojena na výstup číslo 13. Rx a Tx diody signalizují komunikaci přes sériovou linku.
8. 8-bitový procesor ATmega328P. Hlavní řídicí jednotka desky Arduino.
9. Napájecí piny, pomocí kterých napájíme připojené komponenty. Je zde 3,3 V a 5 V pin.
10. Analogové piny. Do těchto vstupů se zapojují komponenty, které měří analogovou veličinu, například teploměr. Piny A4 a A5 se používají také jako piny SDA (data) a SCL (clock – hodiny) pro sběrnici I²C – *inter-integrated circuit*. Sběrnice I²C může propojit až 128 komponent, které jsou připojeny pouze dvěma vodiči. Každé připojené zařízení má svou vlastní přidělenou adresu a díky tomu víme, s kterým zařízením probíhá komunikace.

11. Signalizační LED dioda. Při připojení napájení Arduina svítí zeleně.
12. ICSP hlavice, pomocí které se dá deska externě programovat. Využívají ji některé shieldy.

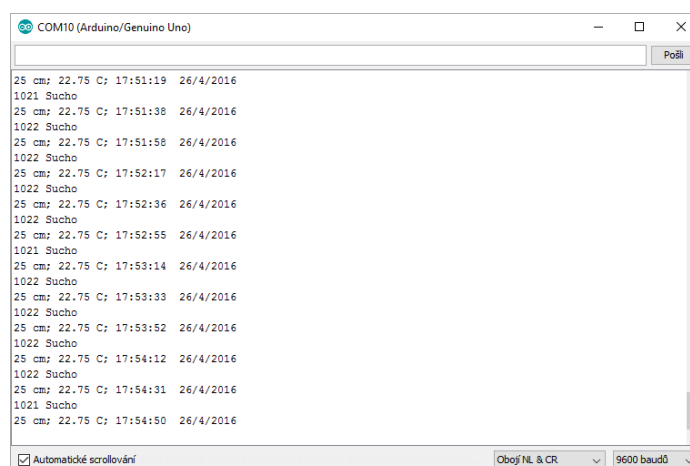
[Voda, 2014]

1.4 Arduino IDE a programování Arduina v knihovně Wiring

Arduino IDE je oficiální integrované vývojové prostředí, díky kterému můžeme Arduino připojit, komunikovat s ním a také vytvářet a nahrávat zdrojový kód. Obsahuje textový editor pro psaní kódu, okno zpráv (například pro chybové hlášky při kompilaci kódu), sériový monitor (například pro čtení právě měřených veličin) (Obrázek 1.5), lištu s ovládacími tlačítky a kontextové menu. [Arduino.cc, 2016]



Obrázek 1.4: Arduino IDE hlavní obrazovka



Obrázek 1.5: Sériový monitor s výpisem hodnot

Arduino se může programovat v jazyku C nebo C++. Pro začínající programátory je vhodnější používat knihovnu Wiring. Knihovna Wiring se používá pro programování Arduina a v současné době je velmi rozšířená.

Po vytvoření nového projektu se nám zobrazí dva bloky programu. První je část *void setup()* sloužící pro zapsání kódu, který se po zapnutí zařízení provede pouze jednou. Druhou částí je *void loop()*, ve kterém se nachází část kódu, která se bude neustále opakovat až do odpojení nebo vypnutí Arduina. Obě tyto části musí program obsahovat, i přesto že by neobsahovaly žádný kód. Pomocí lomítek se píše komentáře, do nichž můžeme napsat různé poznámky týkající se průběhu programu, nebo jen odstavit část kódu. Tento text nemá žádný vliv na funkčnost programu, slouží pouze pro jeho zpřehlednění. [Voda, 2014]

2 Možnosti automatizovaných závlahových systémů

Jsou různé způsoby, jak vytvořit automatizovaný závlahový systém. Všechno závisí na konkrétních požadavcích a (okolních) podmínkách. Využití automatizovaného zavlažování má spoustu výhod, ke kterým patří například úspora času, možnost závlahy v nočních hodinách, rostlina vždy dostane potřebnou tekutinu.

Na výběr je dnes z mnoha automatických závlahových systémů, které se liší způsobem použití, provedením a v neposlední řadě také cenou. Cena se liší podle toho, jak velkou oblast chceme zavlažovat. Jednou z nejlevnějších možností je využití vývojové desky Arduina, pro vytvoření vlastního systému. Ovšem tato možnost není nejjednodušší a je spíše určena pro kutily, kteří se nebojí výzev a mají zájem se něco nového přiučit. Ale díky velkému rozšiřování použití Arduina už můžeme na webu najít několik hotových projektů na téma zavlažování, které můžeme taky použít. V dalších kapitolách se seznámíme s možnými variantami těchto systémů.

2.1 Způsoby zavlažování

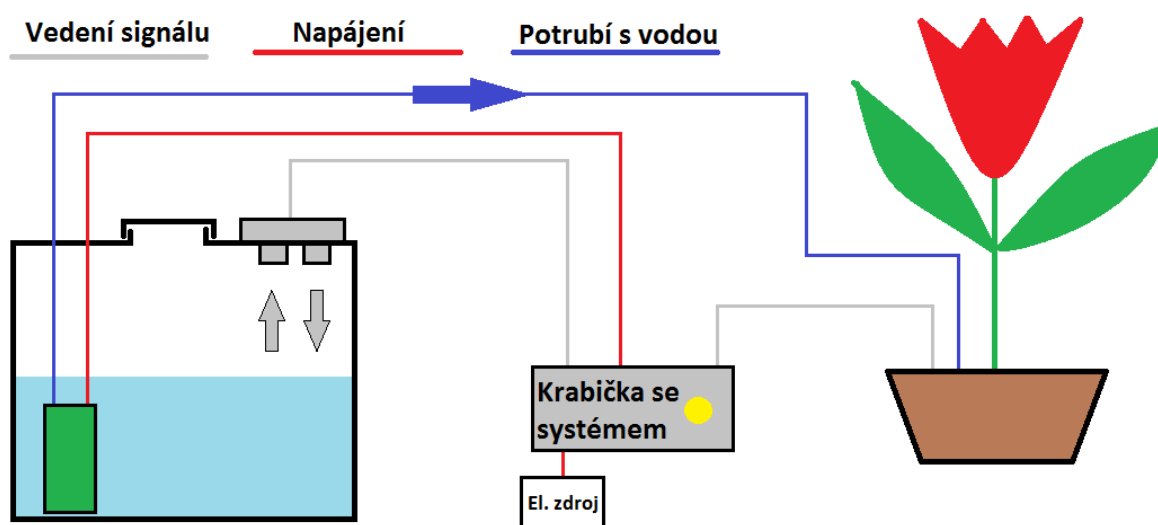
Prvním způsobem zavlažování je zavlažování podpovrchové, tzn. přivádět vodu ke kořenům rostlin pomocí kanálů, nebo podzemním potrubím. Tekoucí vodu tak neregistrujeme a povrch půdy není rozmočený. Nevýhodou je možnost ucpání potrubí.

Druhým způsobem může být zavlažování postřikem. Z vodního zdroje dopravíme tekutinu potrubím k oblasti určené pro zavlažování, kde je rozstřikovači rozstřikována na danou oblast. Můžeme používat zavlažovače, které se otáčí o 360 stupňů a zavlažují víceméně kruhovou oblast, nebo natažením hadice s otvory nad zavlažovanou oblastí. Pro tento způsob musíme přivádět kapalinu pod vyšším tlakem. Nevýhodou je usazování nečistot a ucpání rozstřikovacích otvorů.

Dalším způsobem je zavlažování lokalizované. Tento způsob využiji u svého systému. Je jednoduché a spočívá v tom, že tekutinu přivedeme k rostlině pomocí hadice nebo potrubí.

2.2 Popis fungování zavlažovacího systému

Můj systém bude využívat způsob lokalizovaného zavlažování rostliny v květináči. Pomocí senzoru vlhkosti budu snímat vlhkost půdy v květináči a při poklesu vlhkosti pod danou hodnotu se zapne čerpadlo a přivede do květináče dané množství vody. Systém také bude obsahovat teploměr, který bude zaznamenávat okolní teplotu. Ten poslouží mimo jiné pro vytvoření grafu okolní teploty a vlhkosti půdy.



Obrázek 2.1: Zjednodušené schéma zavlažovacího systému

Jak již bylo zmíněno, pro přívod vody použijí hadici, která bude mít na konci ponorné čerpadlo. Jako zdroj vody použijí nádrž. Místo čerpadla by se dal použít ventil. V tomto případě je nutné zajistit, aby v hadici byl dostatečný tlak pro rozvod tekutiny, například umístěním nádrže s vodou do vyšší pozice, než je vyústění hadice. V některých systémech se využívá senzor světla, který slouží pro měření intenzity světla. Dá se použít do grafů teploty a vlhkosti půdy.

Pro zobrazování aktuálních hodnot teploty, vlhkosti půdy atd. můžeme k systému připojit LCD displej, na kterém se tyto veličiny budou zobrazovat.

Závlahový systém pracuje automaticky, můžeme jej ale ovládat i manuálně, kdy k systému připojíme ovládací tlačítka, nebo jej ovládáme pomocí počítače.

3 Výběr komponent

Základní součástí je již popisované Arduino UNO, které bude sloužit jako řídicí jednotka. Pro měření teploty jsem vybral vodotěsný digitální teploměr DS18B20 (Obrázek 3.1). Teploměr má teplotní rozsah $-55\text{ }^{\circ}\text{C}$ až $+125\text{ }^{\circ}\text{C}$, které převádí pomocí 9 – 12 bitového Analog-Digital převodníku s rozlišením $0,0625\text{ }^{\circ}\text{C}$ při využití



Obrázek 3.1: Vodotěsný digitální teploměr [Elec Freaks, 2016]

12 bitů. Teploměr má tři vodiče, jeden z nich je GND – zem, sloužící pro uzemnění obvodu, dalším je VDD – který slouží pro napájení 3 – 5,5 V a třetím je DQ – digitální vstup/výstup, tento vodič lze použít i v parazitním napájecím módu. Tento mód umožňuje teploměru fungovat bez využití VDD vodiče, čili externího napájení, protože využívá napájení pomocí DQ vodiče, když je sběrnice nabitá. Pro uchování proudu, když je sběrnice vybitá, se používá

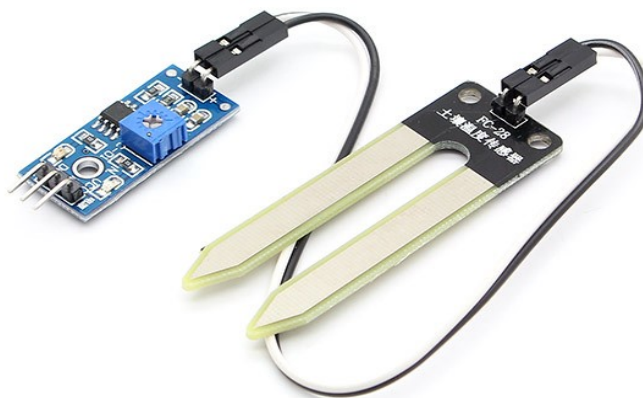
kondenzátor určený pro parazitní napájení. Při tomto módu musí být vodič externího napájení VDD uzemněn. [Maxim Integrated, 2016]

Pro měření vlhkosti půdy jsem vybral senzor vlhkosti půdy (Obrázek 3.3), který má analogový nebo digitální výstup. Digitální výstup je nastavitelný připojeným potenciometrem. Senzor má také tři vodiče, jeden pro napájení 3,3 – 5 V, další pro uzemnění a třetí je digitální výstup z převodníku nebo analogový výstup přímo ze senzoru vlhkosti.

Pro přívod vody jsem vybral vodní elektrické ponorné čerpadlo COMET. Pracovní napětí čerpadla je 12 V. Průtok čerpadla je 8 litrů za minutu, což bude bohatě stačit. Pro ovládání čerpadla použiji 5 V jednobokové relé, které má budicí proud 15 - 20 mA. Napájení čerpadla obstará 12-ti voltový síťový zdroj.



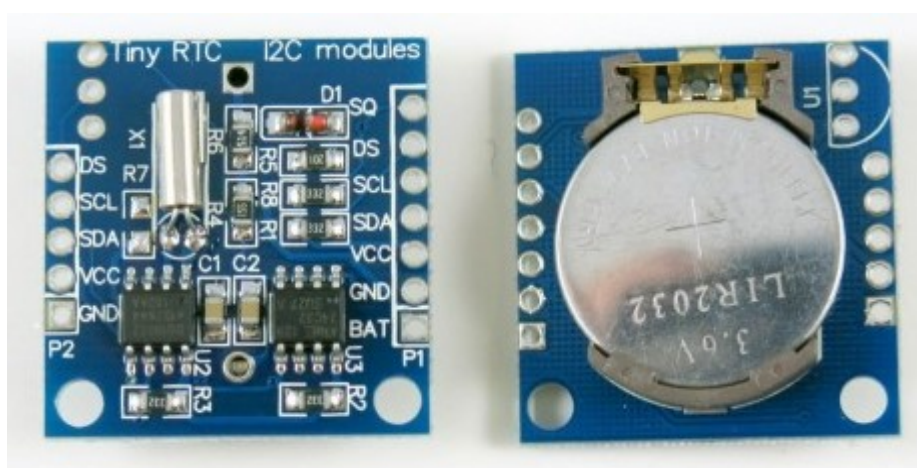
Obrázek 3.2: Vodní elektrické ponorné čerpadlo [Jacksons, 2016]



Obrázek 3.3: Senzor vlhkosti půdy [Santy, 2016]

Pro měření výšky hladiny jsem zvolil senzor HC-SR04. Jeho rozsah měření je od 2 do 400 cm s přesností měření 0,3 cm a pracovní napětí je 5 V. Princip jeho fungování spočívá v tom, že senzor vyšle ultrazvukový signál a měří čas, dokud se signál vrátí zpět. Senzor má čtyři vodiče, jeden slouží pro jeho napájení, druhý pro uzemnění, třetí vysílá signál a čtvrtý přijímá signál.

Použiji také modul hodin reálného času DS1307 (RTC). Tento modul slouží k uchování aktuálního času a data. Díky tomu, že umožňuje vložení vlastní 3 V CR2032 baterie, se čas po odpojení, nebo resetu Arduina, nevynuluje a nebude se muset pokaždé ručně znovu zadávat. Modul komunikuje s Arduinem pomocí I²C sběrnice. [Maxim Integrated, 2016]



Obrázek 3.4: Hodiny reálného času DS1307 [Mikroelectron, 2016]

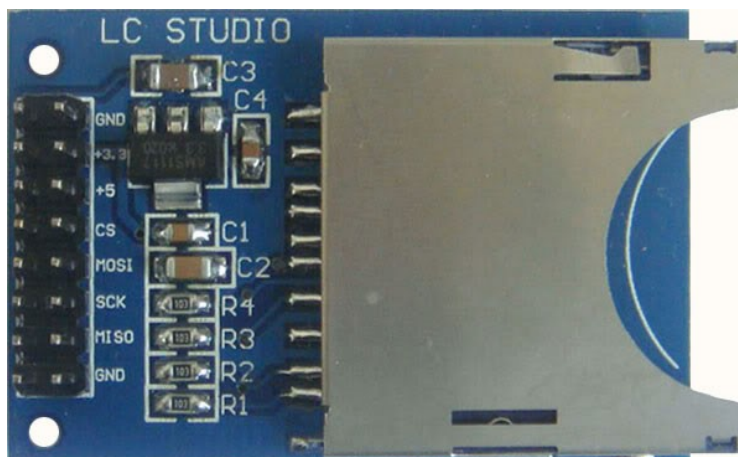


Obrázek 3.5: Ultrazvukový senzor vzdálenosti HC-SR04

Pro komunikaci s počítačem a odesílání získaných dat, jsem zvolil TCP/IP WiFi modul ESP8266 s integrovanou anténou. Tento modul je levný a komunikuje pomocí standardního

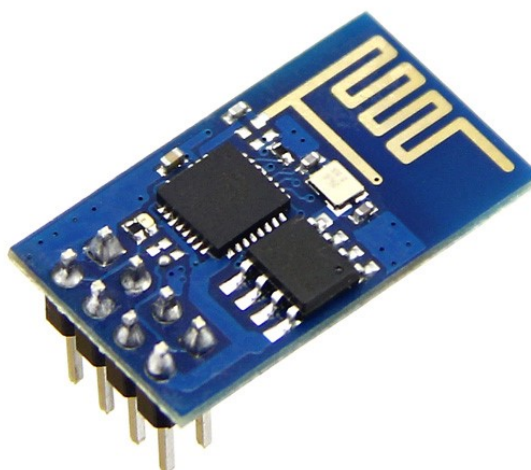
WiFi protokolu 802.11 b/g/n. Modul komunikuje s Arduinem přes SPI sběrnici a jeho napájecí napětí je 3,3 V. [ESP8266 Serial WIFI Module, 2016]

K ukládání naměřených dat poslouží modul s čtečkou SD karet, která komunikuje s Arduinem pomocí SPI sběrnice. Modul je napájen 3,3 V nebo 5 V. Pro ukládání dat je dostačující 2 GB SD karta.



Obrázek 3.6: Čtečka SD karet

Pro napájení Arduina použijí 12 V síťový zdroj. Také budeme potřebovat LED pro signalizaci prázdné nádrže, vodiče pro propojení komponent s Arduinem a materiál pro vytvoření obalu, ve kterém bude Arduino.



Obrázek 3.7: TCP/IP WiFi modul ESP8266 [Seedstudio, ESP8266]

Tabulka 3.8: Cena jednotlivých komponentů.

Komponent:	Cena: [Kč]
Arduino UNO	436
Teploměr DS18B20	90
Ultrazvukový měřič vzdálenosti HC-SR04	85
Hodiny reálného času DS1307	61
Relé 5 V	73
WiFi modul ESP8266	194
Čtečka SD karet	47
Síťový zdroj 12 V, 0,5 A, 6 W	200
Vodní čerpadlo	240
Síťový zdroj 12 V, 2 A, 24 W	109
Senzor vlhkosti	100
LED	3
SD karta	100
Krabička	65
Nádrž, hadice	Dle požadavků
Vodiče, podložky, šrouby, matice	70

Celková cena těchto zakoupených komponent je 1873 Kč.

4 Návrh systému

Pro zajištění automatické závlahy rostliny budeme potřebovat řídicí jednotku Arduino s výše popsanými moduly a senzory. Senzor vlhkosti půdy bude uložen v květináči a přes analogový pin připojen k Arduino. Jakmile se hodnota vlhkosti půdy dostane pod určenou hodnotu, která bude nastavena v programu dle vlastních požadavků a bude signalizovat nutnost zavlažení půdy, vyšle se z Arduina signál, který sepne relé připojené na digitálním výstupu a tím se spustí čerpadlo. Ponorné čerpadlo bude uloženo v nádrži s vodou. Čerpadlo nesmí být delší dobu zapnuto naprázdno, tzn. aniž by čerpalo vodu. K tomuto nám bude sloužit ultrazvukový senzor vzdálenosti, připojený k Arduino pomocí dvou vodičů do digitálního vstupu a výstupu, který bude připevněn na horní straně nádrže. Ten bude měřit výšku hladiny, a jakmile bude v nádrži spodní kritická hranice vody, vypne a zakáže spuštění čerpadla. Také se rozsvítí LED umístěná na krabičce se systémem, ta bude signalizovat nutnost doplnění vody do nádrže. Po doplnění nádrže, bude systém moci zase zavlažovat. Dále v systému použiji čtečku SD karet, která slouží i pro zápis na SD kartu. Na SD kartu se budou zapisovat naměřené hodnoty ze senzorů (teplota a vlhkost). A abych mohl zaznamenat, v jakém čase byly konkrétní hodnoty naměřeny, připojím k Arduino modul reálného času, který bude časový údaj také ukládat na SD kartu. Tyto data se budou posílat na webovou stránku pomocí připojeného WiFi modulu, kde budou zobrazeny ve formě grafů.

4.1 Možné technické problémy

Díky tomu, že už existuje několik variací zavlažovacích systémů s deskou Arduino, jsem se dozvěděl o několika problémech, které by mohly nastat při realizaci mého systému.

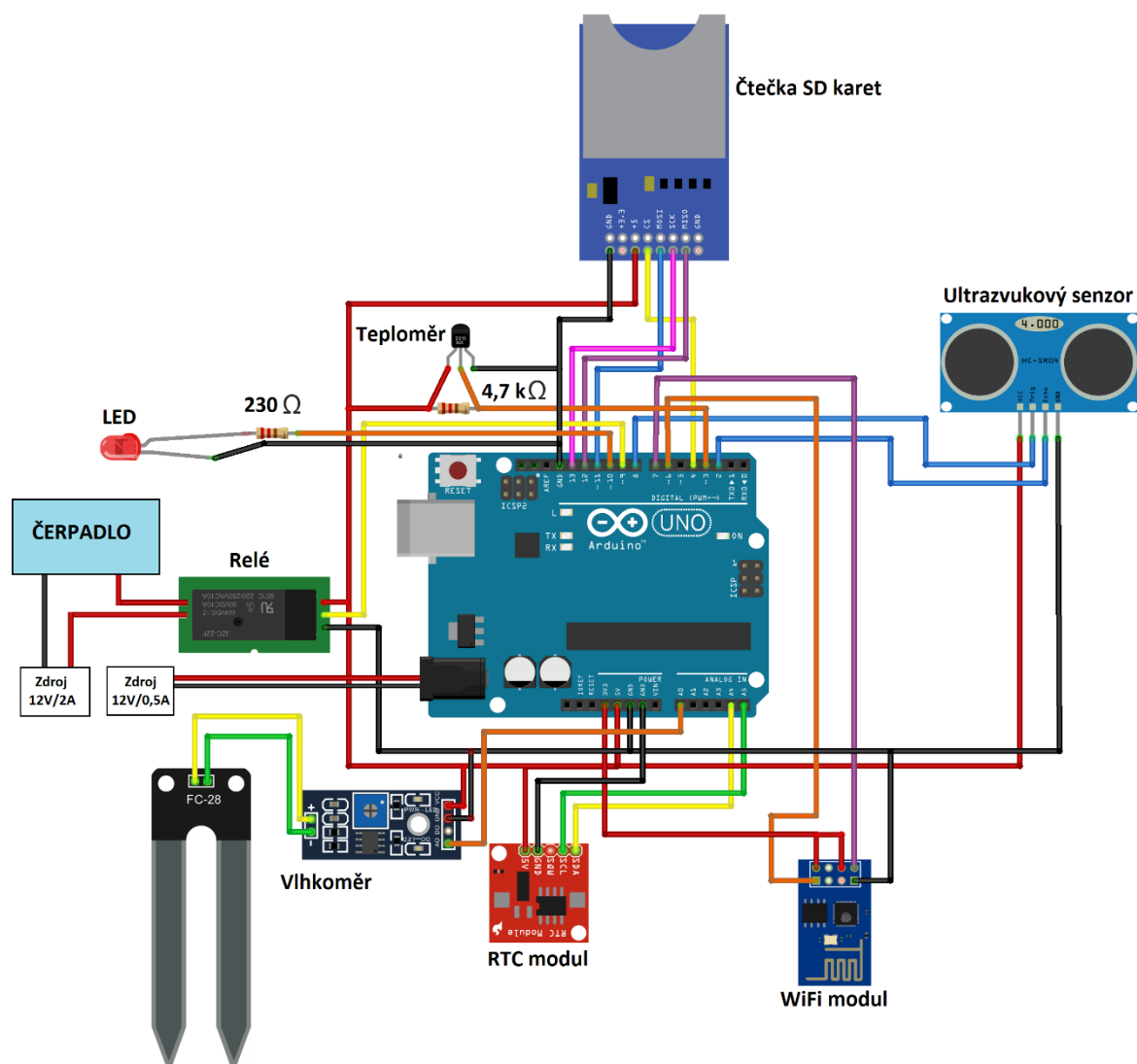
Prvním možným problémem je správnost měření výšky hladiny pomocí ultrazvukového senzoru. Pro nejpresnější měření bude nutné dodržet, aby byl senzor co nejvíc kolmý k hladině vody v nádrži. [Pihrt, 2016]

Také kvůli toho, že má Arduino pouze omezenou RAM paměť a modul pro WiFi a čtečka SD karet tuto paměť dost zaplňují, se může stát, že nebudou najednou oba moduly fungovat. Také zprávy `Serial.println(““)`, které vypisují hodnoty, se ukládají do paměti RAM. Toto lze ošetřit tím, že se tyto zprávy budou místo do RAM paměti ukládat do FLASH

paměti Arduino pomocí příkazu `Serial.println(F(““))`. Využití paměti RAM se také dá snížit správným výběrem datového typu jednotlivých hodnot. [Margolis, 2012]

4.2 Schéma zapojení systému

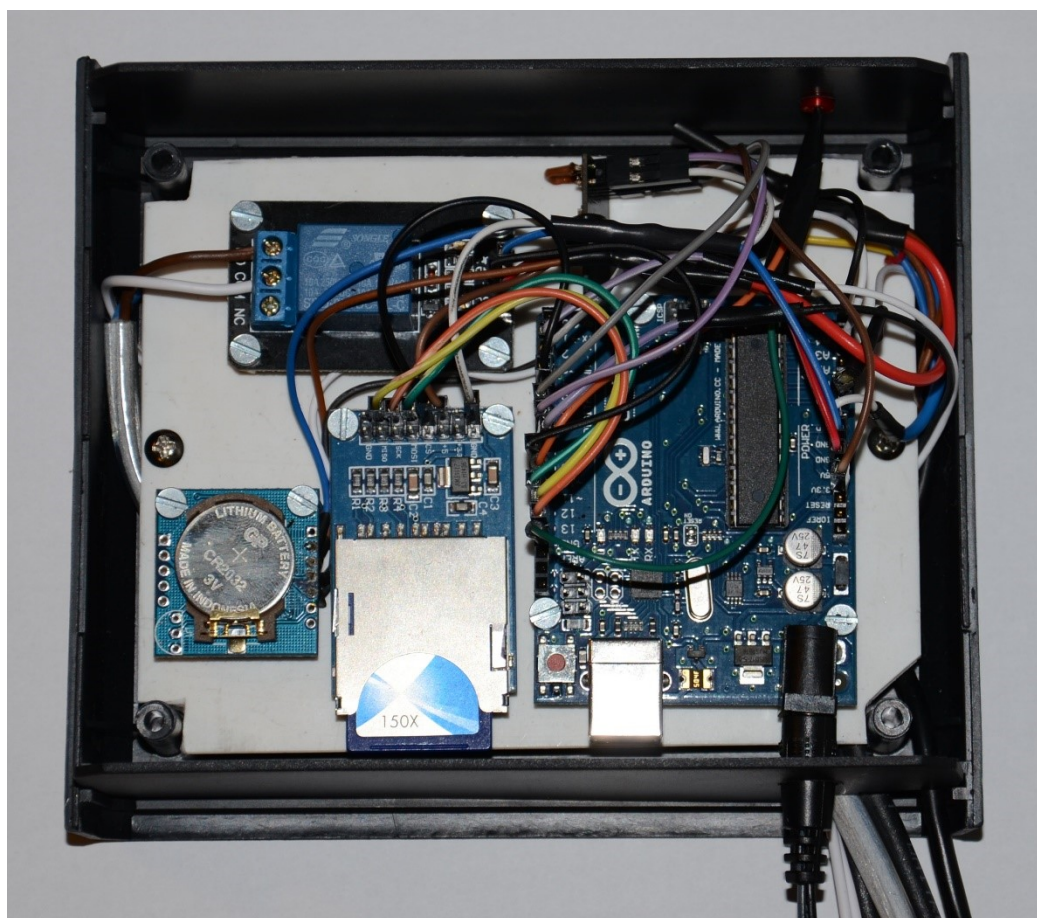
Schéma je vytvořeno v programu Fritzing. Červenou barvou je vyznačen napájecí vodič, černou barvou vodič uzemnění. Barevně jsou vyznačeny vodiče sloužící pro přenos dat a komunikaci.



Obrázek 4.1: Schéma zapojení systému

4.3 Vytvoření systému

V první řadě bylo nutné obstarat obal, krabičku, ve kterém bude celý systém uložen. Zvolil jsem plastovou krabičku o rozměrech 15x13x5 cm. Do ní jsem přišrouboval plastovou desku, na které je pomocí distančních podložek připevněno Arduino, modul reálného času, čtečka SD karet, relé a WiFi modul. Tyto komponenty jsou zapojeny vodiči podle elektrického schématu. Některé vodiče lze jednoduše nasadit na pin komponenty a druhou stranou vložit do vstupu Arduina, některé bylo nutno připájet natrvalo.



Obrázek 4.2: Uložení komponent v krabičce

Na zadní straně krabičky je vyřezán otvor pro vývod napájecího kabelu Arduina i ponorného čerpadla, kabelu teploměru, senzoru vlhkosti a ultrazvukového senzoru.

Čerpadlo je uloženo v nádrži, která je vytvořena z kanystru o objemu 0,005 m³, kde je na horní straně udělán otvor pro ultrazvukový snímač vzdálenosti.

4.4 Připojení komponent k Arduinu

Každá komponenta používá jiný způsob komunikace s Arduinem, tudíž má každá vlastní způsob zapojení. Jako první jsem zapojil komponenty, které budou spolu s Arduinem v krabičce.

Zapojení RTC modulu. Tento modul pro komunikaci využívá I²C sběrnici. Ta funguje na principu *master-slave* komunikace. To znamená, že Arduino – *master* zahajuje a ukončuje komunikaci s RTC modulem – *slave*. Komunikace probíhá pomocí dvou vodičů: SCL (generuje hodinový signál) a SDA (datový signál). Na desce Arduina slouží pro komunikaci pomocí I²C piny A4(SDA) a A5(SCL). Jako záložní zdroj energie modul využívá slot pro připojení 3 V CR 3032 baterie.

Modul SD karty komunikuje přes SPI sběrnici a obsahuje 4 vodiče, SCLK (generuje hodinový signál), MOSI (*Master Out, Slave In*), MISO (*Master In, Slave Out*) a SS (*Slave Select*). Tyto vodiče jsou připojeny k pinům A4 (SS), A11 (MOSI), A12 (MISO), A13 (SCLK). V modulu je vložena 2 GB karta.

Relé je připojeno k Arduinu třemi vodiči. První IN1 je připojen k pinu D7 a slouží pro ovládání relátka další je vodič napájení (VCC) a uzemnění (GND). Na spínacích kontaktech relé je připojen vodič +12 V ze zdroje a vodič z čerpadla. Vodič 0 V ze zdroje je přímo propojen s druhým vodičem čerpadla.

WiFi modul obsahuje 8 pinů a 5 z nich je připojeno k Arduinu. Jsou to piny Rx (*Receive*) sloužící pro příjem dat a je připojen na pin D6, Tx (*Transmit*) sloužící pro posílání dat připojen na D7, pin pro napájení, které je 3,3 V a pin pro uzemnění. Zbylé 3 piny na modulu slouží pro jeho programování a přehrávání firmwaru.

U teploměru DS18B20 musíme připojit rezistor o velikosti 4.7 k Ω mezi vodič datový a napájecí. Datový vodič je připojen k pinu D8 na Arduinu.

Ultrazvukový měřič vzdálenosti HC SR-04 potřebuje pro komunikaci s Arduinem dva digitální piny, jeden je určen jako spouštěč ultrazvukového signálu a je připojen na pin D3 a druhý jako jeho zpětná vazba připojen na pin D2.

Senzor vlhkosti může být připojen dvěma způsoby. Jeden využívá digitálního pinu a vysílá signál pouze, pokud je překročena nastavená hranice vlhkosti na potenciometru. Druhý využívá analogového pinu a měří úroveň vlhkosti kontinuálně. Já využiji druhého způsobu a připojím senzor k pinu A0.

Červená LED je umístěna na přední straně krabičky a je připojena vodičem, na kterém je ještě připojen rezistor o odporu $330\ \Omega$, na pin D10. Druhým vodičem je připojena na zem. Její napájení je tak zajištěno z digitálního výstupu.

Všechny komponenty potřebují pro napájení 5 V (kromě WiFi modulu, ten potřebuje pouze 3,3 V). Na Arduinu se nachází pouze jeden pin sloužící pro napájení 5 V a dva piny pro uzemnění. Tudíž je třeba všechny potřebné napájecí vodiče připájet k sobě a následně připojit k Arduinu. To stejné je potřeba udělat s vodičem pro uzemnění.



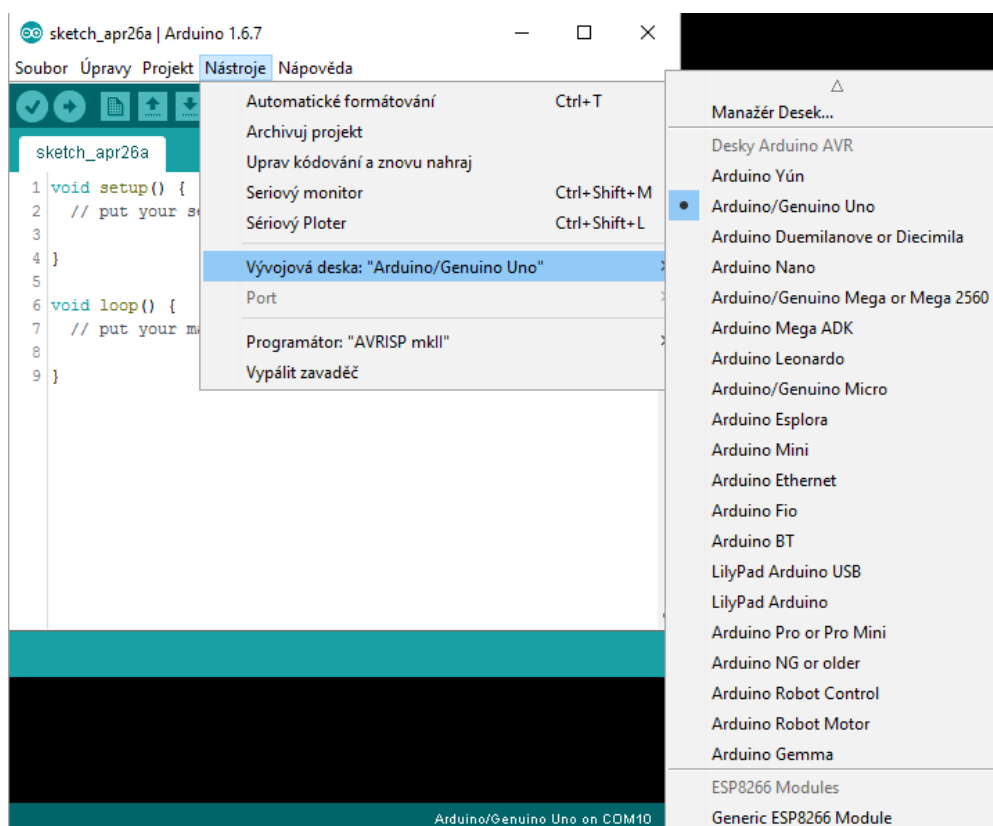
Obrázek 4.3: Automatický závlahový systém

5 Vytváření programu

Pro vytvoření programu použijí již zmíněné Arduino IDE. Toto vývojové prostředí má prvky usnadňující programování Arduina, některé z nich si popíšeme níže.

5.1 Nastavení IDE

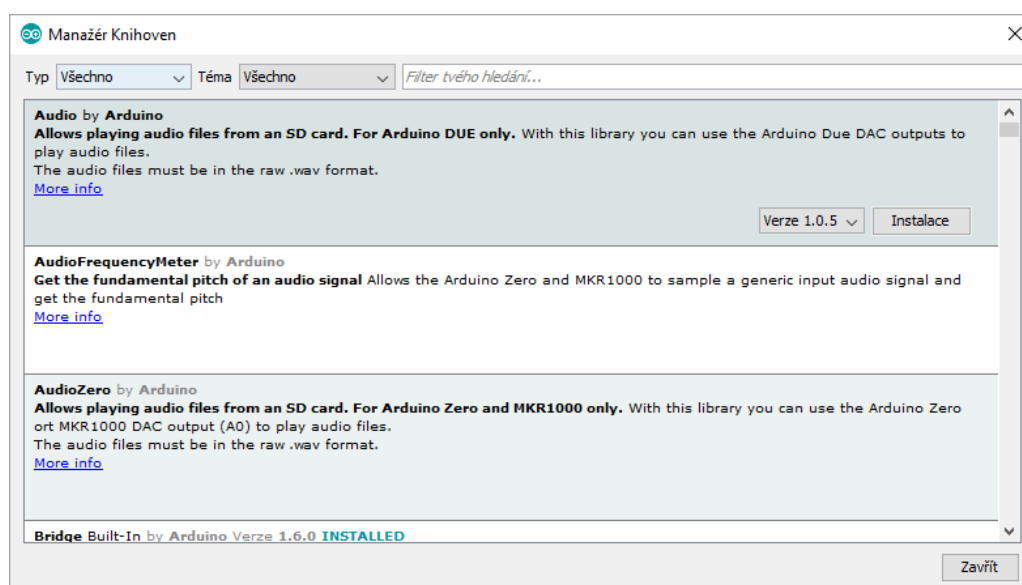
Nejprve je nutné si správně nastavit vývojové prostředí. Je třeba zvolit správnou vývojovou desku v *Nástroje* → *Vývojová deska*, aby program například věděl, které piny je možno použít, případně jaké jsou další možnosti použité desky. Dále je třeba specifikovat port, na kterém je připojeno Arduino, abychom s ním mohli začít komunikovat. Teď již můžeme začít vytvářet program a nahrávat jej do Arduina.



Obrázek 5.1: Výběr vývojové desky a portu

5.2 Vložení knihoven a definice proměnných

Na začátku programovacího kódu se vkládají potřebné knihovny. Knihovny jsou soubory funkcí a procedur. Obsahují již hotový kód pro použití/ovládání komponent nebo sběrnic. Arduino IDE již v základu obsahuje kolem dvaceti knihoven, další se dají přidat pomocí správce knihoven (Obrázek 5.2) v menu *Projekt->Přidat knihovnu->Spravovat knihovny*, kde si můžeme stáhnout i jiné oficiální knihovny. Na webu existují stovky dalších knihoven, které v tomto správci nenajdeme, tyto knihovny se proto musí přidat ručně.



Obrázek 5.2: Správce knihoven

V mém kódu jsem použil sedm knihoven. Knihovny se do kódu přidávají příkazem `#include`. Knihovna `DS1307.h` obsahuje kód pro ovládání modulu reálných hodin, tento modul komunikuje přes sběrnici I²C, kterou nám umožní použít knihovna `Wire.h`. Knihovny `SPI.h` a `SD.h` umožňují používat sběrnici SPI a modul čtečky SD karet. Knihovna `OneWire.h` slouží pro komunikaci s digitálním teploměrem, ten také využívá knihovny `DallasTemperature.h`. [Studt, 2007] Jako poslední je knihovna `SoftwareSerial.h`. Tato knihovna umožňuje nastavit sériovou komunikaci na kterýkoliv jiný digitální pin, než jsou původní piny D0 a D1. Tento krok jsem zvolil, jelikož při připojení WiFi modulu, který využívá sériové komunikace na původních pinech, jsem nemohl nahrávat program do Arduina.

V další části kódu je vhodné nadefinovat konstanty jako čísla pinů připojených komponent.

```
Const char ECHOPIN 2      // Echo pin z HC-SC04 na pin D2
Const char TRIGPIN 8      // Trig pin z HC-SC04 na pin D8
Const char ONE_WIRE_BUS 3 // Teploměr na pinu D3
Const char VLHKOMER A0    // Vlhkoměr na pinu A0
Const char RELE 9         // Relé na pinu D9
Const char LED 10         // LED na pinu D10
```

Dále se definují zbylé konstanty a příkazy.

```
String apiKey = "YO27HPEQDKX6M139"; // Thingspeak write API key
SoftwareSerial ser(7, 6); // Definování pinů nové sér. kom. RX, TX
OneWire ourWire(ONE_WIRE_BUS); // Nastaví oneWire komunikaci
DallasTemperature sensors(&ourWire); // Nastaví použití oneWire knihovny
char chipSelect = 4; // Definuje SD modul na pinu D4
File mySensorData; // Definuje soubor
```

5.3 Program pro zavlažování

Nyní se dostáváme k hlavnímu bodu programu, který je rozdělen na dvě části. V první se nachází funkce *void setup()* v níž je definován začátek sériové komunikace a spuštění knihovny pro digitální teploměr. Dále jsou definovány použité digitální piny pomocí funkce *pinMode()* podle toho, zdali jsou vstupní, nebo výstupní.

Při prvotním sestavení a spuštění programu je zde kód pro nastavení času modulu RTC. Po jeho nastavení a nahrání do Arduina, se musí tento kód vymazat, nebo dát do poznámek, protože jinak by se při každém spuštění systému znovu nastavoval čas. Díky baterii v modulu, se tento čas uchová i po odpojení systému z elektrické sítě.

Tato část proběhne pouze jednou a to při zapojení systému do elektrické sítě.

V části *void loop()* je uložena část kódu, která se neustále opakuje. Na začátku je kód pro zjištění výšky hladiny v nádrži. Funkcí *digitalWrite(TRIGPIN, HIGH)* se vyšle impuls a pomocí funkce *pulseIn(ECHOPIN, HIGH)* se změří odezva impulsů. Vzdálenost se následně vypočítá z rychlosti zvuku, která je při teplotě vzduchu 25 °C $346,3 \text{ m}\cdot\text{s}^{-1}$, tj. $0,03463 \text{ cm}\cdot\mu\text{s}^{-1}$. Jelikož se signál odráží, je nutno rychlost podělit dvěma a výsledná konstanta 0,017315 se vynásobí s dobou odezvy, takto získáme výšku hladiny.

Vlhkoměr měří vlhkost na základě změny elektrického odporu v půdě. Je připojen k analogovému vstupu a hodnota vlhkosti je v hodnotách od 0 do 1023, kdy 0 značí nejvyšší vlhkost a 1023 zase nejvyšší sucho. Tuto hodnoty přečteme příkazem *analogRead(VLHKOMER)* a následně vypíšeme do sériového monitoru příkazem *Serial.print(vlhkost)*. V následující části je obsažena podmínka, která určuje, kdy má být čerpadlo zapnuto a kdy se má rozsvítit LED signalizující nedostatek vody v nádrži. Po změření vlhkosti a zjištění, že je půda nedostatečně vlhká se na sériový port vypíše „Sucho“ a čerpadlo začne čerpat vodu po dobu 3 s, což by mělo být cca. 0,4 litru vody. Pokud je půda dostatečně vlhká, vypíše se na sériový monitor status „Vlhko“ a program bude pokračovat dál. Aby se čerpadlo nezničilo, jakmile by se snažilo čerpat vodu i přesto, že by v nádrži nebyla voda, je připsána podmínka, kdy vzdálenost hladiny od senzoru vzdálenosti nesmí být větší než 20 cm. Tato hodnota je nastavena podle výšky nádrže, tak aby sací část čerpadla byla vždy pod vodou. V tomto případě se čerpadlo nezapne a pro signalizaci prázdné nádrže se rozsvítí červená LED na přední straně krabičky se systémem. Kontrola stavu hladiny včetně rozsvícení LED je i za stavu, kdy je půda dostatečně vlhká. Mohlo by se totiž stát, že by byla hladina vody dostatečně vysoká, tudíž by systém půdu zavlažil, ale přitom by hladina klesla pod kritickou hranici a v příštím snímání by sice půda byla vlhká, ale nevěděli bychom o nutnosti doplnění vody v nádrži, dokud by se půda znovu nevysušila.

```
// Vlhkoměr
int vlhkost = analogRead(VLHKOMER); // Změří vlhkost
Serial.print(vlhkost);           //Hodnota 0 max. vlhko, 1023 min. vlhko
if (vlhkost >= 800) {
    Serial.println(F(" Sucho"));
    if (vzdalenost >= 20) {
        digitalWrite(LED, HIGH); // Rozsvítí LED
    }
    else {
        digitalWrite(LED, LOW);
        digitalWrite(RELE, LOW); // Zapne čerpadlo
        delay(3000);
        digitalWrite(RELE, HIGH); // Vypne čerpadlo
    }
}
if (vlhkost < 800) {
```

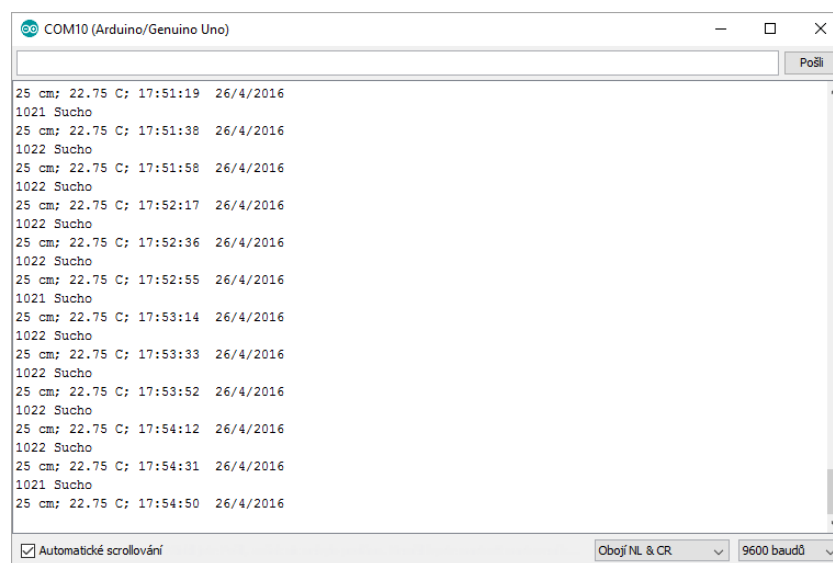
```

Serial.println(F(" Vlhko"));
if (vzdalenost >= 20) {
    digitalWrite(LED, HIGH); // Rozsvítí LED
}
else {
    digitalWrite(LED, LOW); // Zhasne LED
}
}

```

Použité hodnoty, jako je hranice vlhkosti kdy je potřeba půdu zavlažit, doba zapnutého čerpadla a kritická hladina vody v nádrži se odvíjí od vlastních požadavků, použitého čerpadla a nádrže.

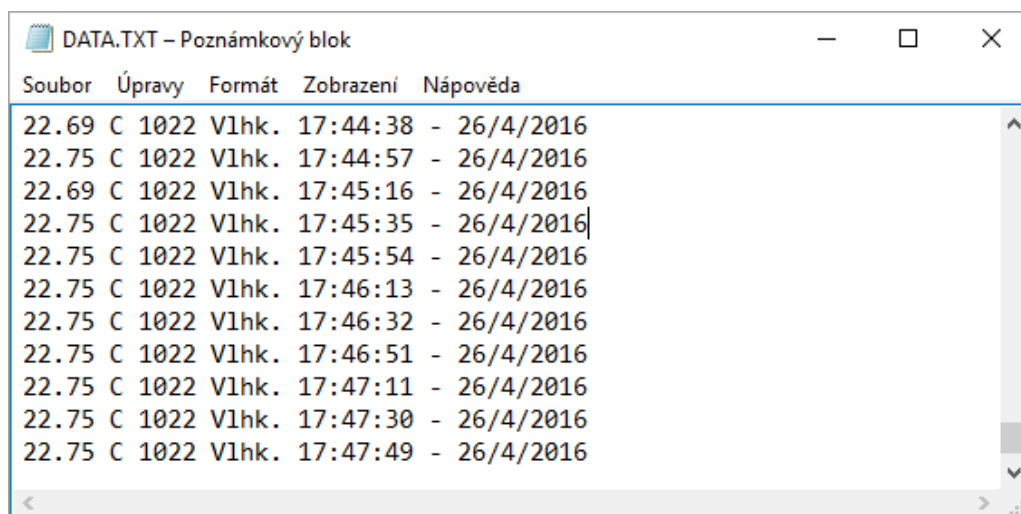
Příkaz `sensors.requestTemperatures()` nám vyšle požadavek na zjištění teploty. Příkazem `float teplota = sensors.getTempCByIndex(0)` uložíme zjištěnou teplotu do konstanty `teplota` s přesností na dvě desetinné místa.



Obrázek 5.3: Výpis dat v serial monitoru

Ukládání dat na SD kartu je dáno příkazem `mySensorData = SD.open("Data.txt", FILE_WRITE)`, který buď vytvoří, nebo pokud již je vytvořen tak otevře soubor `Data.txt`, do kterého se naměřené data budou ukládat. Pod tímto příkazem je podmínka přítomnosti SD karty a vytvoření souboru, která když je splněna, tak nám vypíše pomocí sérií příkazů `Serial.print()` vzdálenost mezi hladinou a senzorem vzdálenosti, teplotu z teploměru

a aktuální čas s datem. Na kartu se poté příkazem *mySensorData.print()* uloží teplota, vlhkost a aktuální čas. Nakonec se soubor uzavře příkazem *mySensorData.close()*.



Obrázek 5.4: Soubor na SD kartě s uloženými daty

5.4 Připojení ESP8266 k WiFi síti

WiFi modul ESP8266 používá vlastní AT příkazy, pomocí kterých s modulem komunikujeme a nastavujeme jej. Komunikace probíhá v sériovém monitoru vývojového prostředí. Po zapojení modulu je třeba otevřít sériový monitor, nastavit dole v možnostech „Obojí NL & CR“ a rychlost přenosu na 115200 baudů, jelikož jsou tyto moduly defaultně nastaveny na tuto rychlost. Moduly s novějším firmware podporují příkaz pro změnu modulační rychlosti. Nejprve otestujeme, jestli můžeme s modulem komunikovat napsáním příkazu *AT*. Pokud se vrátí odpověď *OK*, je vše v pořádku, pokud se vrátí nesmyslné znaky, je třeba změnit rychlost v nastavení serial monitoru. Běžně používaná rychlost u Arduina je 9600Bd, tímto příkazem tedy nastavíme požadovanou rychlost *AT+UART=9600,8,1,0*, kde 9600 značí rychlost v baudech, 8 je počet přenášených bitů, 1 značí existenci paritního bitu a 0 značí počet stop bitů. Poté tedy přepneme modulační rychlost i v sériovém monitoru a pokračujeme příkazem *AT+CWMODE=1*, který nastaví WiFi modul do režimu Sta, což je mód routeru. Příkazem *AT+CWMUX=1* nastavíme vícenásobné připojení a příkazem *AT+CWLAP=“SSID“,“PASSWORD“* po dosazení názvu sítě a hesla se modul připojí. Příkazem *AT+CIFSR* je možno vypsat přidělenou adresu modulu. [ESP8266 Serial WIFI Module, 2016]

Jelikož je nastavení WiFi sítě jednorázová věc vytvořil jsem program, který po vyplnění názvu a hesla sítě ke které se chceme připojit, modul sám připojí. Také z důvodu, že v programu pro zavlažování používám SoftwareSerial knihovnu a mám tedy modul připojen k jiným než defaultním pinům pro sériovou komunikaci, je třeba příkazy posílat na vytvořenou SoftwareSerial linku. Sériový monitor totiž posílá příkazy pouze na defaultní sériovou linku.

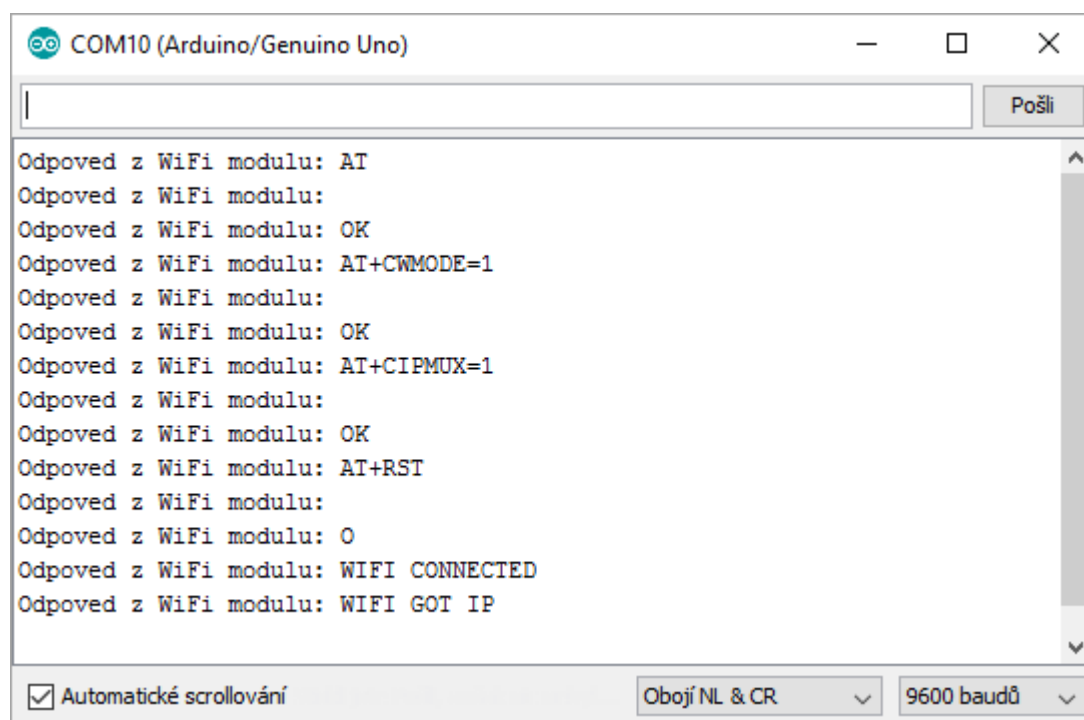
```
#include <SoftwareSerial.h>

SoftwareSerial wifi(7, 6);           //Rx,Tx pin

void setup() {
    Serial.begin(9600);
    wifi.begin(9600);
    delay(1000);
    wifi.println("AT");
    delay(1000);
    wifi.println("AT+CWMODE=1");
    delay(1000);
    wifi.println("AT+CIPMUX=1");
    delay(1000);
    wifi.println("AT+RST");
    delay(1000);
    wifi.println("AT+CWJAP=ssid,password"); //SSID,PASSWORD
    delay(1000);

    // wifi.println("AT+CWQAP"); // odpojeni z Wifi site
    // delay(1000);
}

void loop() {
    while (wifi.available()) {
        String odpoved = wifi.readStringUntil('\n');
        Serial.println("Odpoved z WiFi modulu: " + odpoved);
    }
}
```



Obrázek 5.5: Připojení ESP8266 k WiFi síti

5.5 Posílání dat na ThingSpeak

ThingSpeak je webová služba, sloužící pro ukládání a zpracování dat. Pro její používání je nutno si vytvořit účet na www.thingspeak.com, kde je nutno vyplnit vlastní ID, email a heslo. Po přihlášení se vytvoří *Channel*, ve kterém se budou zobrazovat naměřené hodnoty. V poli *Field 1* bude zobrazena teplota a v poli *Field 2* bude zobrazena hodnota vlhkosti (Obrázek 5.6). Dole je ještě možnost zvolit *Make public*, abychom mohli naměřené hodnoty sledovat i bez nutnosti přihlášení. Po vytvoření kanálu najdeme v záložce *API keys* klíč (Write API key), který budeme potřebovat pro zapisování dat.

Ještě než spustíme TCP připojení, je nutno teplotu a vlhkost převést na datový typ *string*. Poté se program připojí na server ThingSpeak, kam se pomocí příkazů

```
String getStr = "GET /update?api_key=";
getStr += apiKey;
getStr += "&field1=";
getStr += String(strTep);
getStr += "&field2=";
getStr += String(strVlh);
getStr += "\r\n\r\n";
```

připraví data pro odeslání a příkazem *AT+CIPSEND* odešlou přes WiFi na ThingSpeak. ThingSpeak přijímá data v intervalu minimálně 15 s. Z důvodu ustálení vody v půdě a zaznamenání nové hodnoty vlhkosti jsem nakonec programu přidal příkaz *delay(30000)*, který zajistí pauzu programu na 30 s. Naměřená data jsou k dispozici z: <https://thingspeak.com/channels/109851>.

Channel Settings

Percentage complete 70%

Channel ID 109851

Name Zavlažování

Description Záznam teploty a vlhkosti z automatického zavlažovacího systému.

Field 1 Teplota ☒

Field 2 Vlhkost ☒

Field 3 ☐

Field 4 ☐

Field 5 ☐

Field 6 ☐

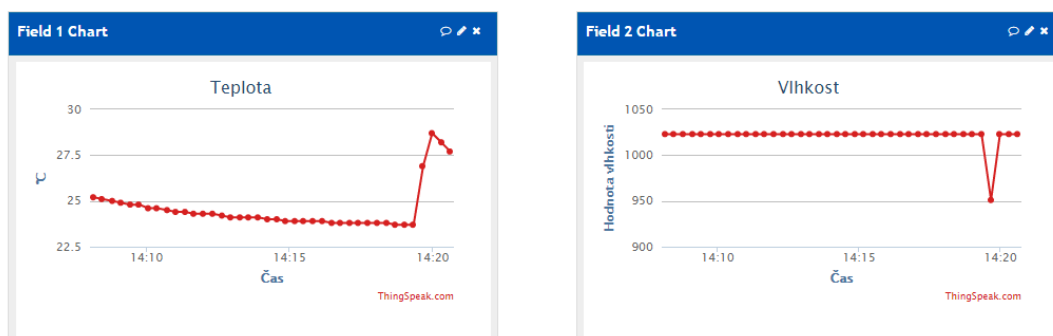
Field 7 ☐

Field 8 ☐

Obrázek 5.6: Nastavení kanálu na ThingSpeak.com

Channel Stats

Created 7 days ago
 Updated less than a minute ago
 Last Entry less than a minute ago
 1170 Entries



Obrázek 5.7: Naměřená data na stránce ThingSpeak

6 Závěr

Cílem této práce bylo vytvořit automatizovaný závlahový systém s logováním dat. Při vytváření systému jsme se seznámili s vývojovou deskou Arduino UNO, která je použita jako řídicí jednotka systému a jejími možnostmi. Dále pak s možnostmi a způsoby zavlažování a návrhem vlastního systému pro zavlažování, včetně analýzy vhodných použitých komponent.

Vývojová deska Arduino UNO je velmi rozšířený open-source hardware. Díky tomu lze na webu najít stovky nejrozličnějších projektů, od blikání s LED po řízení autonomních robotů. Proto je jednodušší pro každého, kdo má chuť naučit se něco nového včetně dnes stále více populárního programování, sestavit si svůj vlastní projekt. V mém případě jsem si vybral automatizovaný závlahový systém, využívající lokalizované zavlažování, tj. přívodem vody na povrch půdy. V případě suché půdy a tedy nutnosti závlahy vyšle Arduino impuls k sepnutí relé, na kterém je připojeno ponorné čerpadlo umístěné v nádrži s vodou. Toto čerpadlo nesmí pracovat více než 20 s bez vody, jinak by došlo k jeho poškození. Z tohoto důvodu je v nádrži ultrazvukový senzor vzdálenosti, který měří výšku hladiny. Při poklesu hladiny pod nastavenou hranici, zakáže spuštění čerpadla a rozsvítí červenou LED na přední straně krabičky systému, abychom věděli, že je třeba doplnit vodu do nádrže.

Systém také obsahuje čtečku SD karet, modul reálných hodin, vodotěsný teploměr a WiFi modul. Teploměr snímá teplotu (okolní, nebo půdy, záleží na umístění) a ukládá ji spolu s hodnotou vlhkosti a časovým údajem na SD kartu. Modul reálných hodin obsahuje vlastní záložní 3V baterii, tudíž ani po odpojení systému od elektrické sítě, se čas nevynuluje a není jej proto nutné pokaždé znovu nastavovat.

WiFi modul v systému slouží pro posílání teploty a hodnoty vlhkosti přes síť internet na web ThingSpeak. Jeho připojení k síti se provádí samostatným programem. Po připojení automaticky posílá v reálném čase naměřené hodnoty na stránku www.ThingSpeak.com, kde dochází k následnému zpracování dat v podobě grafu. Tato funkce se hodí mimo jiné v případě, kdy si všimneme, že je půda stále suchá a nedochází k zavlažování. To by bylo s největší pravděpodobností způsobeno nedostatkem vody v nádrži. Časové rozmezí grafu se dá nastavit.

Pro přesnější měření vlhkosti, zvláště na větších plochách půdy, by bylo vhodné použít více senzorů vlhkosti. V mém případě závlahy v květináči stačil pouze jeden. Také naměřená data by se mohla posílat do vlastní databáze, ze které by se následně načítala na vytvořené webové stránce. V tomto případě by ale bylo nutné použít jinou vývojovou desku, protože Arduino UNO má velmi omezenou FLASH i RAM paměť a už v mém případě použití jsem musel udělat některá řešení pro optimalizaci a i počet pinů byl skoro plně využit. Dále by se mohl vylepšit obal/krabička systému, který by splňoval větší stupeň krytí, pro použití ve vlhčím prostředí.

Poděkování

Rád bych poděkoval Ing. Pavlu Smutnému, Ph.D. za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování mé bakalářské práce.

7 Použitá literatura

ARDUINO.CC [online]. 2016 [cit. 2016-04-26]. Dostupné z: <https://www.arduino.cc/>

ARDUINO.CZ [online]. 2016 [cit. 2016-04-26]. Dostupné z: <http://arduino.cz/>

CYTRON TECHNOLOGIES. *User's manual HC-SR04*. [online]. [cit. 2015-12-22]. Dostupné z: https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit

ELECFREAKS, *Sensors Temperature* [obrázek]. [online]. [cit. 2016-04-05]. Dostupné z: http://www.electfreaks.com/store/images/Sensors_Temperature_DS18B20_TEM01_02.jpg

ESP8266 *Serial WIFI Module* [online]. 2016 [cit. 2016-04-26]. Dostupné z: http://wiki.iteadstudio.com/ESP8266_Serial_WIFI_Module

FARANA, Radim, SMUTNÝ, Lubomír, VÍTEČEK, Antonín, VÍTEČKOVÁ, Miluše a WAGNEROVÁ, Renata. *Doporučení pro psaní odborných textů z oblasti automatizace a informatiky*. [online]. 2008. Dostupné z: http://www.352.vsb.cz/uc_texty/PsaniOdborText/doporuceni-pro-psani-odbornych-textu-z-oblasti-automatizace-a-informatiky.pdf

JACKSONS, *Cometeleg* [obrázek]. [online]. [cit. 2016-05-04]. Dostupné z: <http://www.jacksons-camping.co.uk/campingimages/pumps/cometeleg.jpg>

MARGOLIS, Michael. *Arduino cookbook*. 2nd ed. Sebastopol, Calif.: O'Reilly, c2012, 699 p. ISBN 1449313876.

MAXIM INTEGRATED. *DS18B20 Datasheet a DS1307 Datasheet*. [online]. [cit. 2015-12-22]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> a <http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>

MELGAR, Enrique Ramos, Ciriaco CASTRO DÍEZ a Przemek JAWORSKI. *Arduino and Kinect projects: Design, build, blow their minds*. New York: Distributed to the book trade worldwide by Springer Science+Business Media, c2012, 393 p. ISBN 9781430241683

MIKROELECTRON, *DS1307* [obrázek]. [online]. [cit. 2016-05-04]. Dostupné z: [http://www.mikroelectron.com/image/cache/data/\\$123-500x500.JPG](http://www.mikroelectron.com/image/cache/data/$123-500x500.JPG)

PIHRT, Martin. *Arduino UNO – automat pro zalévání rostlin*. [online]. [cit. 2016-05-15]. Dostupné z: <https://www.pihrt.com/elektronika/319-arduino-uno-automat-pro-vertikalni-zalevani-rostlin>

SANTY, *Vlhkoměr* [obrázek]. [online]. [cit. 2016-05-04]. Dostupné z: http://www.santy.cz/data/product/132_555.jpg

SEEDSTUDIO, *ESP8266* [obrázek]. [online]. [cit. 2016-05-04]. Dostupné z: <http://www.seeedstudio.com/depot/images/product/WiFi%20Serial%20Transceiver%20Module.jpg>

STUDT, Jim. *OneWire Library* [online]. [cit. 2016-04-26]. Dostupné z: http://www.pjrc.com/teensy/td_libs_OneWire.html

VODA, Zbyšek a TÝM HW KITCHEN. *Průvodce světem Arduina*. [online]. 2014 [cit. 2015-12-08]. Dostupné z: <http://arduino.cz/>